

## Part A Report

Name: **Travis Xie**

### Assignment 6: Perceptron Classification and Training

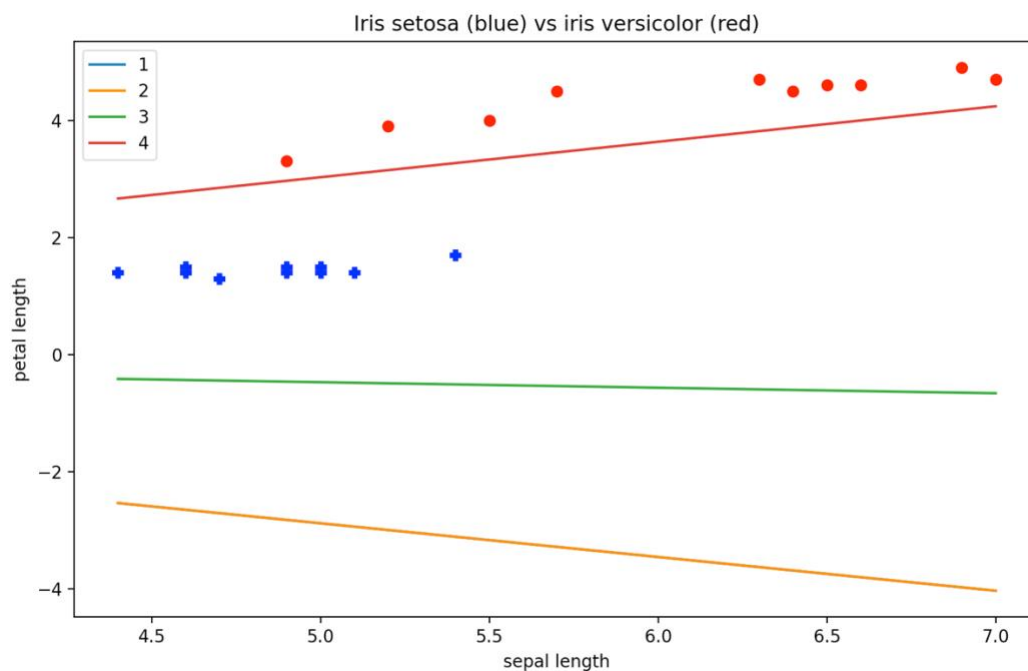
CSE 415 Introduction to Artificial Intelligence, Winter 2021, University of Washington

Please answer each question using text in **Blue**, so your answers stand out from the questions.

QA1. How many epochs were required to train your perceptron on the 2-class Iris data having 2 features?

**4**

QA2. Include a graphic produced using matplotlib that shows both the training data points (in separate colors) and the “separating” lines implied by the weights at the end of each training epoch.” (Reduce the graphic as necessary to make it fit here without taking up more than half the page.)



Was there any thrashing (oscillation in the separator, such as flipping slope back and forth between positive and negative values, or having its y intercept jumping up and down as epochs proceed)? How would you describe the progress of the learning, on the basis of the plot?

**No, there was not any thrashing in the process. The slope of the separator starts as a negative value and then increases to a positive value until it correctly separates the red and blue dots, and so does the y**

intercept. During the process, the agent learns to increase its weight factor to make a correct classification.

QA3. What was the performance of your perceptron on the test data?

Out of 80 testing data, 78 of them are classified correctly, with the correction rate of 97.5%, and 2 are classified incorrectly, with the failure rate of 2.5%.

QA4. After plotting the ring data, describe its distribution in words.

The ring data is equally distributed in all four quadrants, with each quadrant has about 5 red dots and one blue dots. And on each axis ( $x+$ ,  $x-$ ,  $y+$ ,  $y-$ ), there is one blue and one red dot. Besides, the blue dots seem to be enclosed by the red dots.

QA5. Describe the sequence of separators obtained when training your perceptron for 5 epochs using the ring data. To what extent is there convergence? Thrashing? Hope for convergence?

The sequence of separators is from 1 to 5, with 1 being the first separator generated and 5 being the last. The first separator has positive slope and y-intercept, but it is above all the points, no points being separated. So, the next four separators have much smaller y-intercepts that allow them to pass through these points. It seems like that slopes remain positive all the time, but the y-intercepts keep jumping up and down. However, there is no hope for any convergence because the data is not separable by linear lines.

QA6. After you have re-mapped the ring data with the provided non-linear mapping function, plot the data and describe the distribution.

After remapping the data, the points become separable. Their x values distribute over -3 to 3, but all the red points have the y-value greater than 2.5. and all the blue points have the y-value smaller than 2.0.

QA7. After training your perceptron on the re-mapped ring data, did it achieve convergence, and if so, how many epochs were used?

Yes, 15 epochs were used to achieve convergence.

QA8. What does these results suggest about the power of perceptron to classify data that may consist of clusters that cannot be separated by a linear manifold (such as a line or plane)?

The power of perceptron cannot separate data that is not separable.

QA9. Did you run into any difficulties either setting up for Part A or running the programs and answering the questions? If so, please describe them.

At first, I think there is no need to update the bias weight, so I cannot achieve convergence in remapped data because separators are oscillating around  $y = 0$ . But after I update the bias weight the same way I update the other weights, I am able to achieve convergence.

QA10. What portion(s) of Part A did you find most worthwhile and why?

The remapped portion is worthwhile because I can manipulate parameters, like learning rate and the rate of bias weight update, to compare the performance of different agents (the number of epochs they take to converge).